



# **SPRAWOZDANIE**

Grupa projektowa 1

Autor: Michał Deć

Rzeszów, rok 2021

## Spis treści:

1.	Wstęp.....	2
2.	Opis implementacji problemu i podstaw zagadnienia.....	3
3.	Podział programu.....	3
3.1.	Odczyt danych wejściowych.....	3
3.2.	Sortowanie.....	3
3.3.	Zapis wyniku.....	4
4.	Schemat blokowy algorytmu.....	4
5.	Znane ograniczenia programu.....	11
6.	Uruchomienie programu i sprawdzenie wyników.....	11
7.	Złożoność czasowa algorytmu.....	12
8.	Podsumowanie.....	12

## 1. Wstęp

Program ma za zadanie wyszukanie największej możliwej liczby, konstruując ją poprzez złączenie podanych uprzednio liczb.

Przykład:

Wejście: [10, 81, 9]

Wyjście: 98110

## 2. Opis implementacji problemu i podstaw zagadnienia

Należy zauważyć, że aby stworzyć największą możliwą liczbę z uprzednio podanych liczb, to należy te liczby ustawić w takiej kolejności aby każda z kolejnych liczb sąsiadujących ze sobą również dawała największą możliwą wartość.

Przykładowo w tablicy [91, 8, 54, 10] kombinacja 9185410 daje największą możliwą liczbę, jeżeli każda para sąsiadujących ze sobą liczb daje również największą możliwą liczbę:

918 > 891

854 > 548

5410 > 1054

Jeżeli przykładowo w tablicy [7, 9, 10] kombinacja 97 daje większą wartość niż 79, to pozycje tych liczb należy zamienić w tablicy, aby utworzyć [9, 7, 10].

## 3. Podział programu

Program składa się z czterech funkcji, których zastosowanie zostało opisane poniżej:

- void wczytanie\_danych\_wejscowych(void)
- int main()
- int czy\_druga\_liczba\_jest\_bardziej\_znaczaca(int pierwsza\_liczba, int druga\_liczba)
- void zapis\_wyniku(void)

### 3.1. Odczyt danych wejściowych

Wszystkie liczby, z których powinna zostać utworzona największa możliwa liczba powinny zostać zapisane w pliku tekstowym **dane\_wejscowe.txt**. Każda z liczb powinna zostać oddzielona spacją.

Za odczyt danych wejściowych odpowiada funkcja **void wczytanie\_danych\_wejscowych(void)**. Najpierw funkcja ta sprawdza ile liczb jest zapisanych w pliku tekstowym **dane\_wejscowe.txt**, a następnie dokonuje alokacji miejsca na tablice danych, której ilość elementów odpowiada ilości liczb z pliku tekstowego. Następnie funkcja ta odczytuje kolejno liczby z pliku tekstowego i umieszcza je w tablicy danych.

### 3.2. Sortowanie

Za sortowanie liczb w tablicy danych odpowiadają funkcje **int main()** oraz **int czy\_druga\_liczba\_jest\_bardziej\_znaczaca(int pierwsza\_liczba, int druga\_liczba)**.

Za sprawdzenie która kombinacja dwóch sąsiadujących ze sobą liczb daje większą wartość odpowiada funkcja `int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)`, do której przesyłane są kolejno po dwie sąsiadujące ze sobą liczby z tablicy danych. Powyższa funkcja zwraca wartość:

- 0, kiedy kombinacja `<pierwsza_liczba><druga_liczba>` daje większą wartość niż `<druga_liczba><pierwsza_liczba>`, co oznacza że druga liczba nie jest bardziej znacząca,
- 1, w odwrotnej sytuacji, kiedy kombinacja `<druga_liczba><pierwsza_liczba>` daje większą wartość niż `<pierwsza_liczba><druga_liczba>`, czyli w sytuacji kiedy druga liczba jest bardziej znacząca.

Na podstawie zwróconego wyniku funkcja `int main()` podejmuje decyzje czy pozycje dwóch sąsiadujących liczb należy zamienić. Jeśli wynikiem działania funkcji `int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)` jest wartość:

- 0, to dwie sąsiadujące ze sobą liczby należy zostawić na aktualnych pozycjach, gdyż druga z sąsiadujących liczb nie jest bardziej znacząca,
- 1, to pozycje dwóch sąsiadujących ze sobą liczb należy zamienić, gdyż druga liczba jest bardziej znacząca.

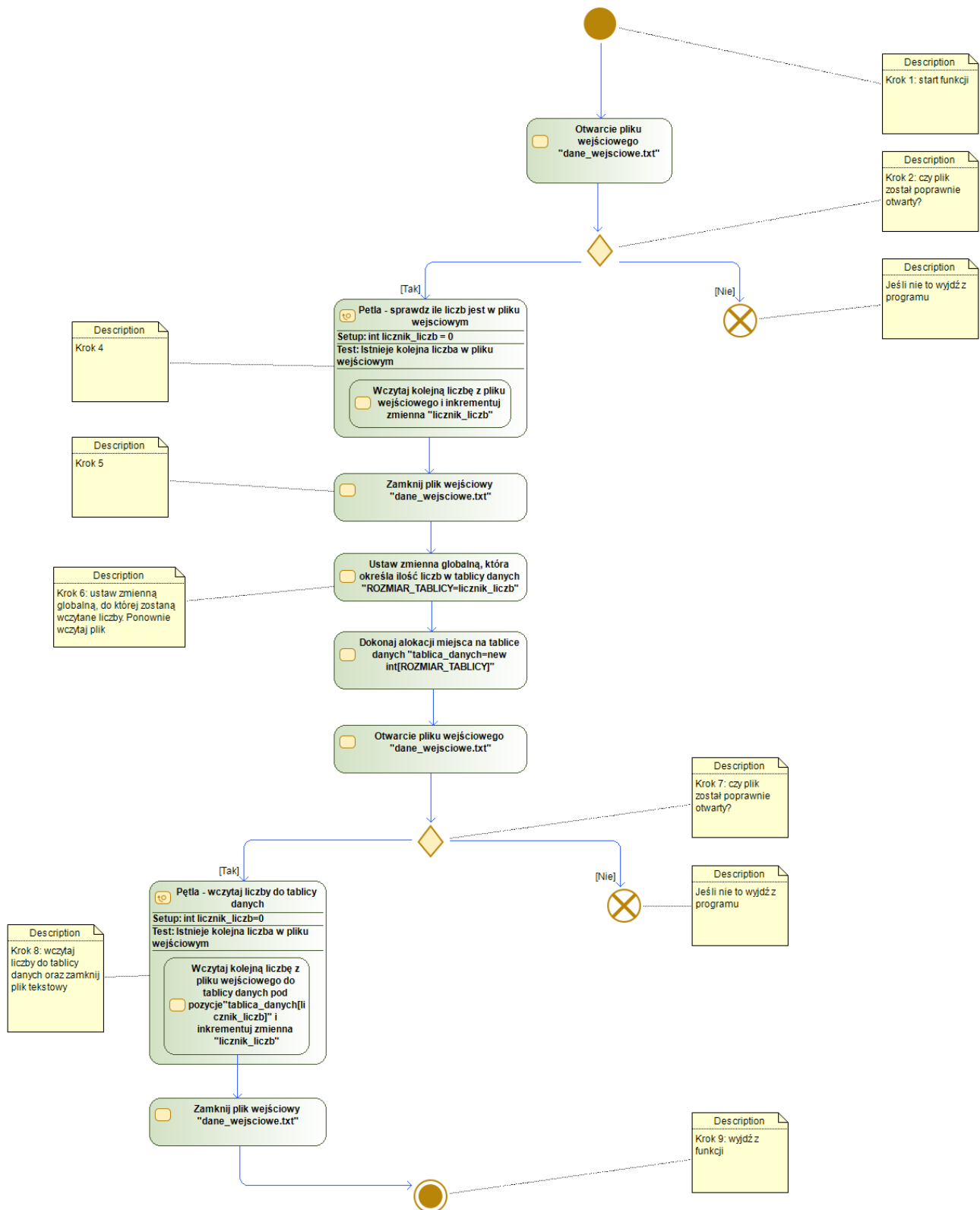
Funkcja `int main()` wywołuje w pętli funkcję `int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)`, przekazując do niej kolejno następne sąsiadujące ze sobą liczby z tablicy danych, do momentu kiedy żadna para sąsiadujących ze sobą liczb w tablicy nie będzie wymagała zamiany pozycji, co oznacza że wszystkie liczby ułożono już na pozycjach, które w wyniku dają największą możliwą liczbę.

### 3.3. Zapis wyniku

Funkcja `void zapis_wyniku(void)` jest odpowiedzialna za zapisanie liczb z posortowanej tablicy danych do pliku wyjściowego `wynik.txt`.

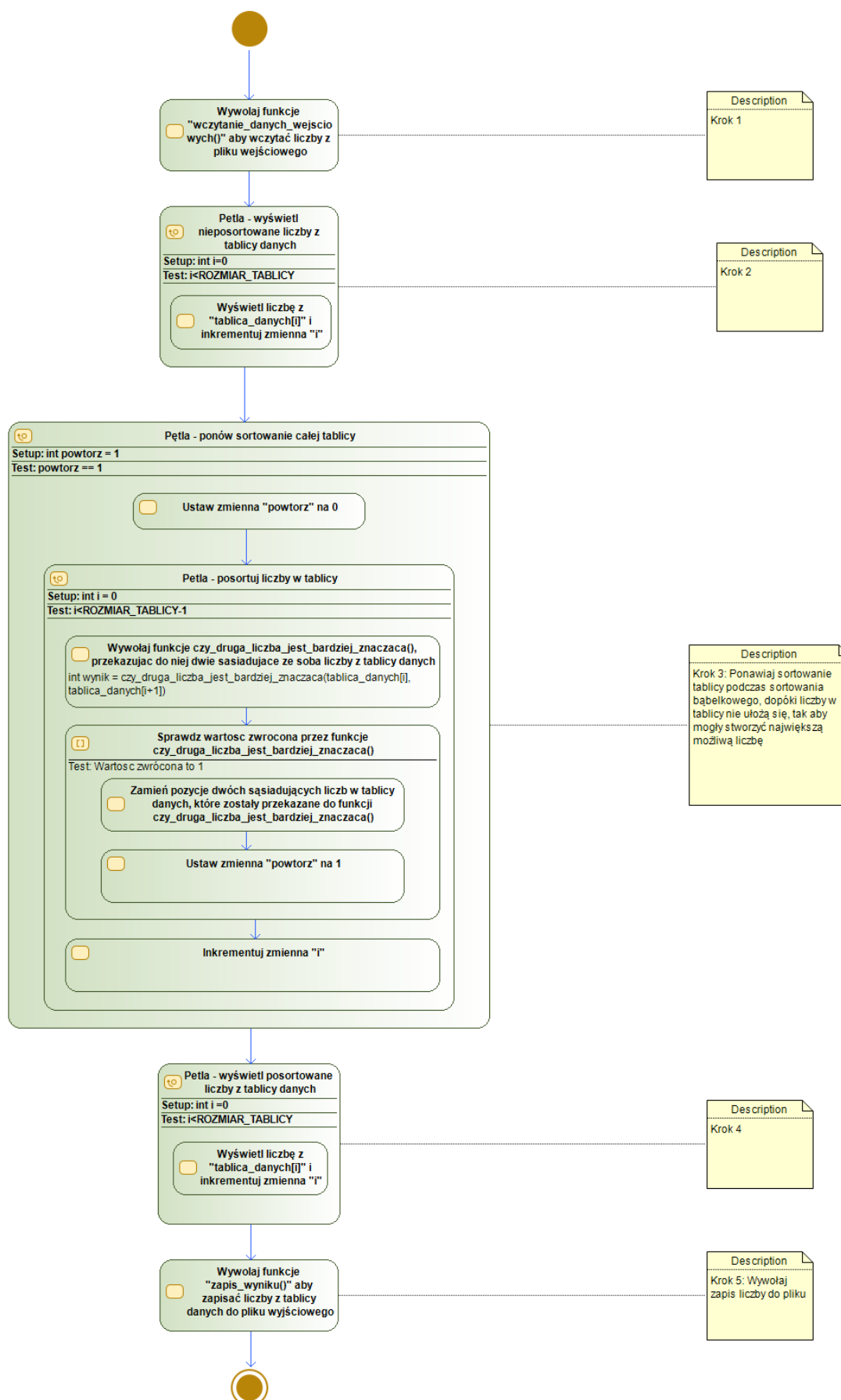
## 4. Schemat blokowy algorytmu

Funkcja void wczytanie\_danych\_wejscowych(void)



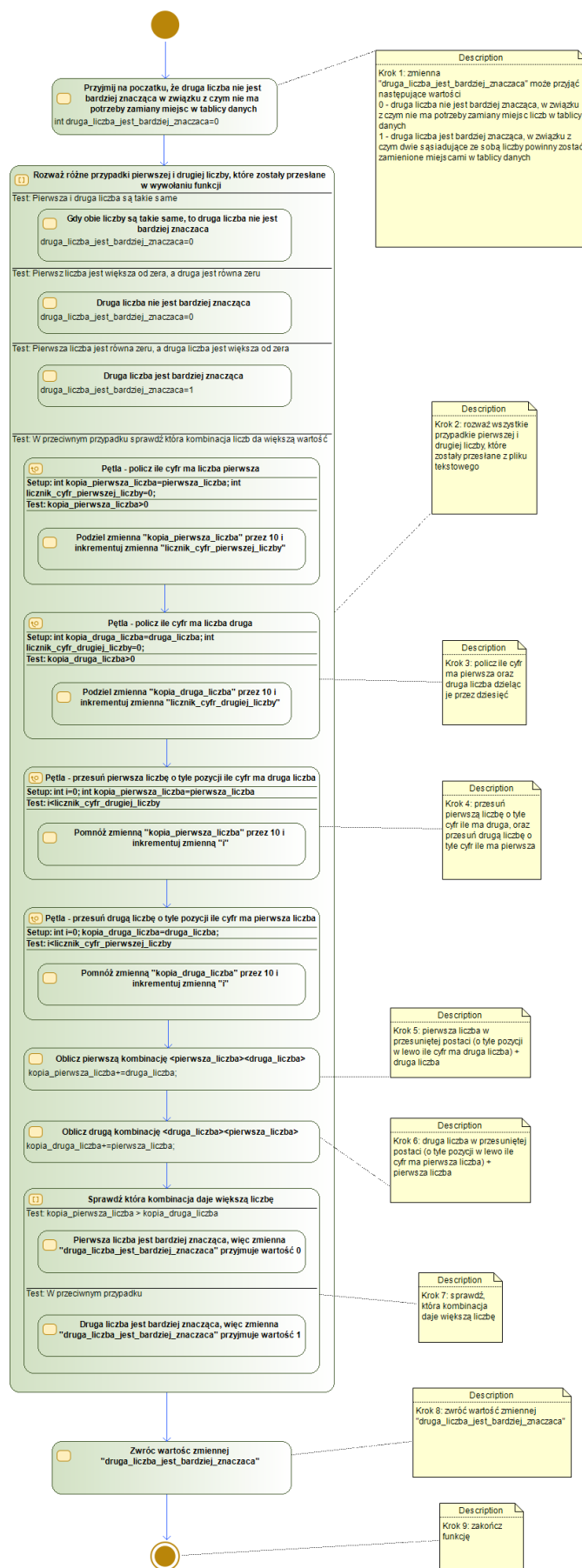


## Funkcja int main()

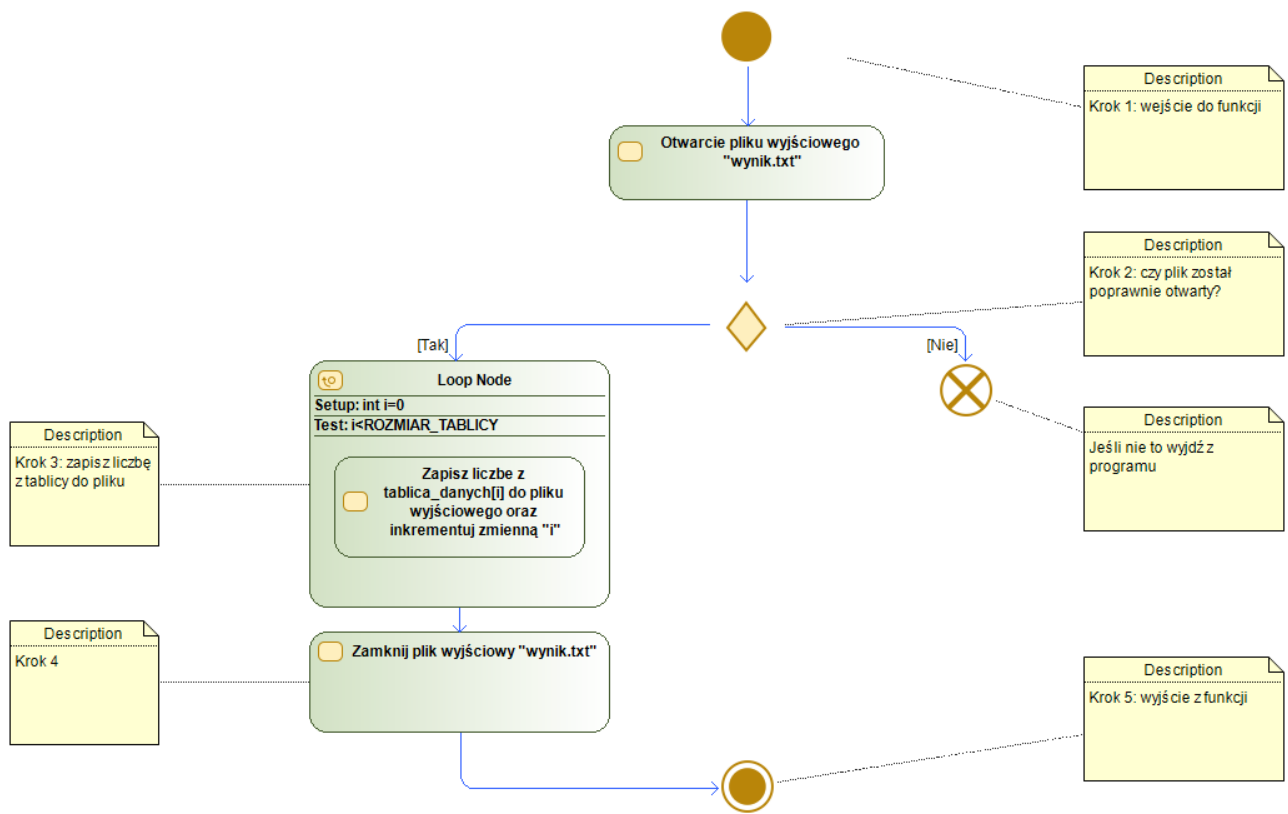




Funkcja int czy\_druga\_liczba\_jest\_bardziej\_znaczaca (int pierwsza\_liczba, int druga\_liczba)



Funkcja void zapis\_wyniku(void)



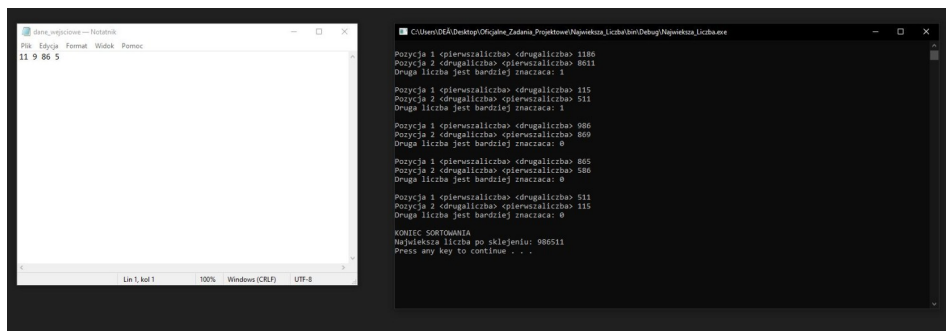
## 5. Znane ograniczenia programu

Program wczytuje liczby z pliku wejściowego do zmiennych typu int oraz oczekuje tylko liczb całkowitych dodatnich lub zera.

## 6. Uruchomienie programu i sprawdzenie wyników

Przykłady testów wykonanych wewnątrz programu:

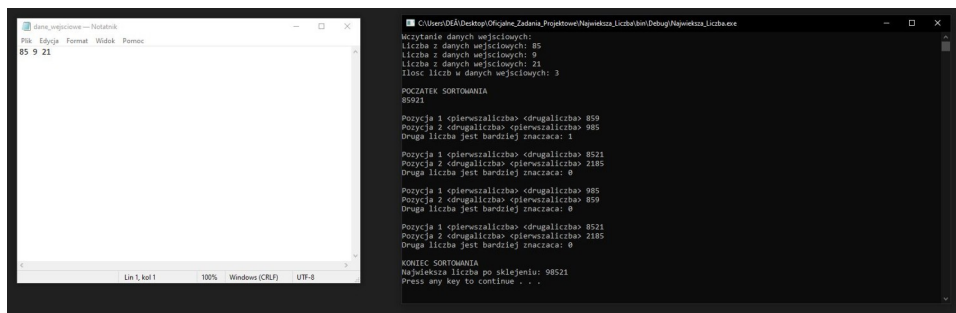
Test 1:



The screenshot shows two windows. On the left is a Notepad++ window titled 'dane wejściowe - Notepad++' containing the text '11 9 86 5'. On the right is a Visual Studio Code debugger window titled 'C:\Users\DEA\Desktop\Oficjalne\_Zadania\_Projektowe\Najwieksza\_Liczba\bin\Debug\Najwieksza\_Liczba.exe'. The output in the debugger window is as follows:

```
Pozycja 1 <pierwszaliczba> <drugaliczba> 1180
Pozycja 2 <drugaliczba> <pierwszaliczba> 8031
Druga liczba jest bardziej znacząca: 1
Pozycja 1 <pierwszaliczba> <drugaliczba> 115
Pozycja 2 <drugaliczba> <pierwszaliczba> 511
Druga liczba jest bardziej znacząca: 1
Pozycja 1 <pierwszaliczba> <drugaliczba> 986
Pozycja 2 <drugaliczba> <pierwszaliczba> 869
Druga liczba jest bardziej znacząca: 0
Pozycja 1 <pierwszaliczba> <drugaliczba> 865
Pozycja 2 <drugaliczba> <pierwszaliczba> 588
Druga liczba jest bardziej znacząca: 0
Pozycja 1 <pierwszaliczba> <drugaliczba> 511
Pozycja 2 <drugaliczba> <pierwszaliczba> 115
Druga liczba jest bardziej znacząca: 0
KONIEC SORTOWANIA
Największa liczba po sklejeniu: 986511
Press any key to continue . . .
```

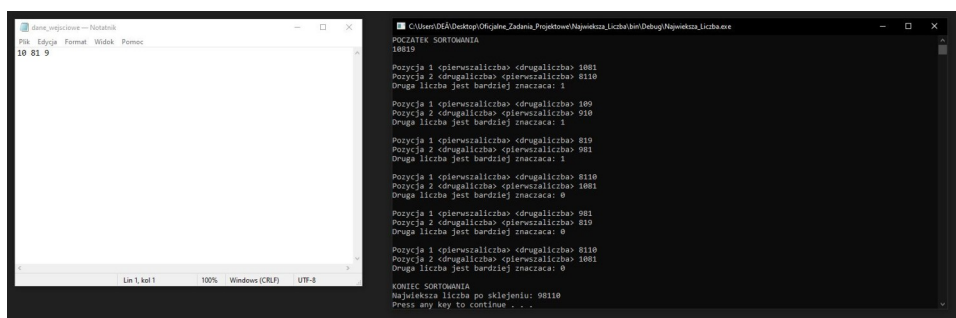
Test 2:



The screenshot shows two windows. On the left is a Notepad++ window titled 'dane wejściowe - Notepad++' containing the text '85 9 21'. On the right is a Visual Studio Code debugger window titled 'C:\Users\DEA\Desktop\Oficjalne\_Zadania\_Projektowe\Najwieksza\_Liczba\bin\Debug\Najwieksza\_Liczba.exe'. The output in the debugger window is as follows:

```
Wczytanie danych wejściowych:
Liczba 1 danych wejściowych: 85
Liczba 2 danych wejściowych: 9
Liczba 3 danych wejściowych: 21
Ilość liczb w danych wejściowych: 3
POCZĄTEK SORTOWANIA
8921
Pozycja 1 <pierwszaliczba> <drugaliczba> 859
Pozycja 2 <drugaliczba> <pierwszaliczba> 985
Druga liczba jest bardziej znacząca: 1
Pozycja 1 <pierwszaliczba> <drugaliczba> 8521
Pozycja 2 <drugaliczba> <pierwszaliczba> 2185
Druga liczba jest bardziej znacząca: 0
Pozycja 1 <pierwszaliczba> <drugaliczba> 985
Pozycja 2 <drugaliczba> <pierwszaliczba> 859
Druga liczba jest bardziej znacząca: 0
Pozycja 1 <pierwszaliczba> <drugaliczba> 8521
Pozycja 2 <drugaliczba> <pierwszaliczba> 2185
Druga liczba jest bardziej znacząca: 0
KONIEC SORTOWANIA
Największa liczba po sklejeniu: 98521
Press any key to continue . . .
```

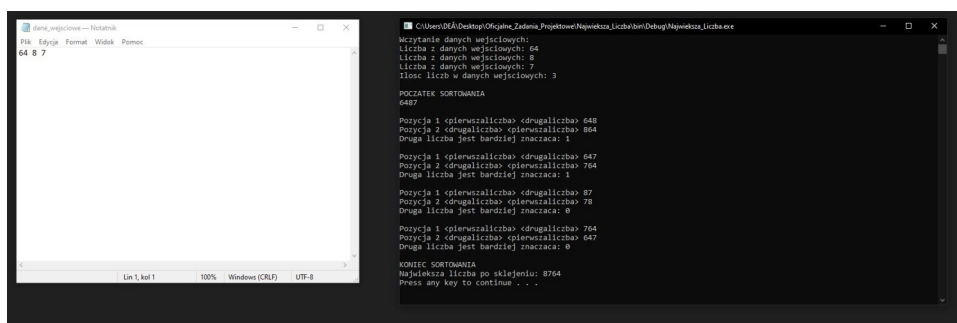
Test 3:



The screenshot shows two windows. On the left is a Notepad++ window titled 'dane wejściowe - Notepad++' containing the text '10 81 9'. On the right is a Visual Studio Code debugger window titled 'C:\Users\DEA\Desktop\Oficjalne\_Zadania\_Projektowe\Najwieksza\_Liczba\bin\Debug\Najwieksza\_Liczba.exe'. The output in the debugger window is as follows:

```
POCZĄTEK SORTOWANIA
18819
Pozycja 1 <pierwszaliczba> <drugaliczba> 1881
Pozycja 2 <drugaliczba> <pierwszaliczba> 8118
Druga liczba jest bardziej znacząca: 1
Pozycja 1 <pierwszaliczba> <drugaliczba> 189
Pozycja 2 <drugaliczba> <pierwszaliczba> 918
Druga liczba jest bardziej znacząca: 1
Pozycja 1 <pierwszaliczba> <drugaliczba> 819
Pozycja 2 <drugaliczba> <pierwszaliczba> 981
Druga liczba jest bardziej znacząca: 1
Pozycja 1 <pierwszaliczba> <drugaliczba> 8118
Pozycja 2 <drugaliczba> <pierwszaliczba> 1081
Druga liczba jest bardziej znacząca: 0
Pozycja 1 <pierwszaliczba> <drugaliczba> 981
Pozycja 2 <drugaliczba> <pierwszaliczba> 819
Druga liczba jest bardziej znacząca: 0
Pozycja 1 <pierwszaliczba> <drugaliczba> 8118
Pozycja 2 <drugaliczba> <pierwszaliczba> 1081
Druga liczba jest bardziej znacząca: 0
KONIEC SORTOWANIA
Największa liczba po sklejeniu: 98118
Press any key to continue . . .
```

## Test 4:



```
Wczytanie danych wejściowych:
Liczba 2 danych wejściowych: 64
Liczba 2 danych wejściowych: 8
Liczba 2 danych wejściowych: 7
Liczba 2 danych wejściowych: 3
POCZĄTEK SORTOWANIA
6487
Porównaj 1 < pierwszaliczba > < drugaliczba > 648
Porównaj 2 < drugaliczba > < pierwszaliczba > 864
Druga liczba jest bardziej znacząca: 1
Porównaj 1 < pierwszaliczba > < drugaliczba > 647
Porównaj 2 < drugaliczba > < pierwszaliczba > 764
Druga liczba jest bardziej znacząca: 1
Porównaj 1 < pierwszaliczba > < drugaliczba > 87
Porównaj 2 < drugaliczba > < pierwszaliczba > 78
Druga liczba jest bardziej znacząca: 0
Porównaj 1 < pierwszaliczba > < drugaliczba > 764
Porównaj 2 < drugaliczba > < pierwszaliczba > 647
Druga liczba jest bardziej znacząca: 0
KONIEC SORTOWANIA
Największa liczba po skłajeniu: 8764
Press any key to continue . . .
```

## 7. Złożoność czasowa algorytmu

Algorytm ma w sobie zagnieżdżoną pętlę w pętli, co czyni go algorytmem o złożoności czasowej  $O(n^2)$ .



## 8. Podsumowanie

Algorytm ma za zadanie sklejać całość z podanych liczb w największą możliwą. Program ten wykorzystuje do tego dzielenie zwykłe przez dziesięć aby porównać cyfry pierwsze oraz sortowanie bąbelkowego do porównania ze sobą kolejnych liczb w tablicy.